MIE334 – Numerical Methods I

Lecture 32: ODEs and Euler Method (C&C 25.1)

Roadmap: **C&C Part Seven**

- Integration of ODEs (**C&C 25**)
  - Euler's Method (25.1)
  - Heun's Method (25.2)
  - Runge-Kutta Methods (25.3)
  - Systems of Equations (25.4)

Review: **Example – Free fall**

- Differential equation:

$$F = ma$$

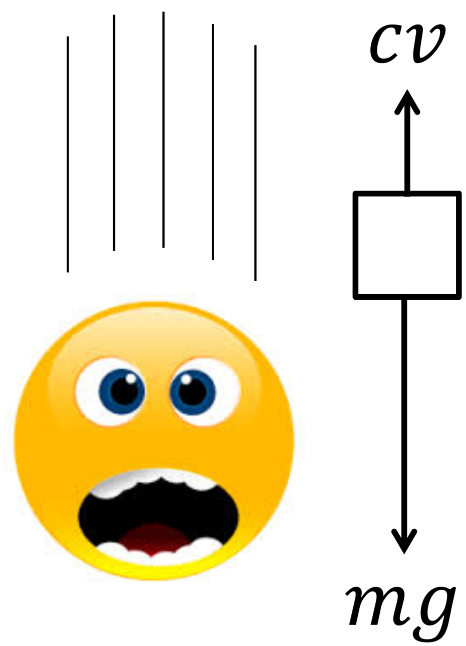
$$mg - cv = m\frac{dv}{dt}$$

- Rewriting:

$$\frac{dv}{dt} = g - \frac{c}{m}v$$

- Solution (for $v = 0$ at $t = 0$):

$$v = \frac{mg}{c}\left(1 - e^{-(c/m)t}\right)$$



Review: **Example – Free fall**

- Application:

  $$v = \frac{mg}{c}\left(1 - e^{-(c/m)t}\right)$$

  - Design c for a prescribed velocity

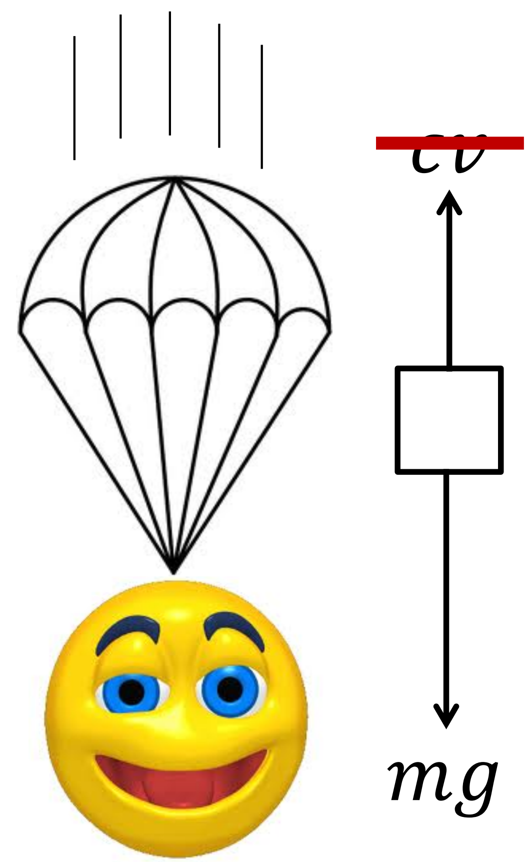
- What if relationship was more complex?

  $$\frac{dv}{dt} = g - \frac{c}{m}\left[v + d\left(\frac{v}{v_{max}}\right)^{e}\right]$$

  - where $d, e, and\ v_{max}$ are empirical constants

- More difficult to solve analytically
  - Need numerical methods to solve $v(t)$

$$c\left[v + d\left(\frac{v}{v_{max}}\right)^{e}\right]$$



Review: **ODEs**

- Ordinary Differential Equations (ODE):
  - Involves one **independent** variable (here, t):

$$\frac{dv}{dt} = g - \frac{c}{m}\left[v + d\left(\frac{v}{v_{max}}\right)^{e}\right]$$

- 1^{st} order: Highest derivative is 1^{st} derivative
- n^{th} order: Highest derivative is n^{th} derivative
- Can convert n^{th} order ODE to n first-order ODEs!
  - Later…

First-order ODEs

- Focus on first-order ODEs of the form:

  $\frac{dy}{dx} = f(x, y)$ or $y' = f(x, y)$

- Goal is to find the original function:

  $y = y(x)$

- Given only the following:

  $y_0 = y(x_0)$

- First-order ODEs only have one **boundary condition**
  - Sometimes called **initial condition**, especially for problems in time

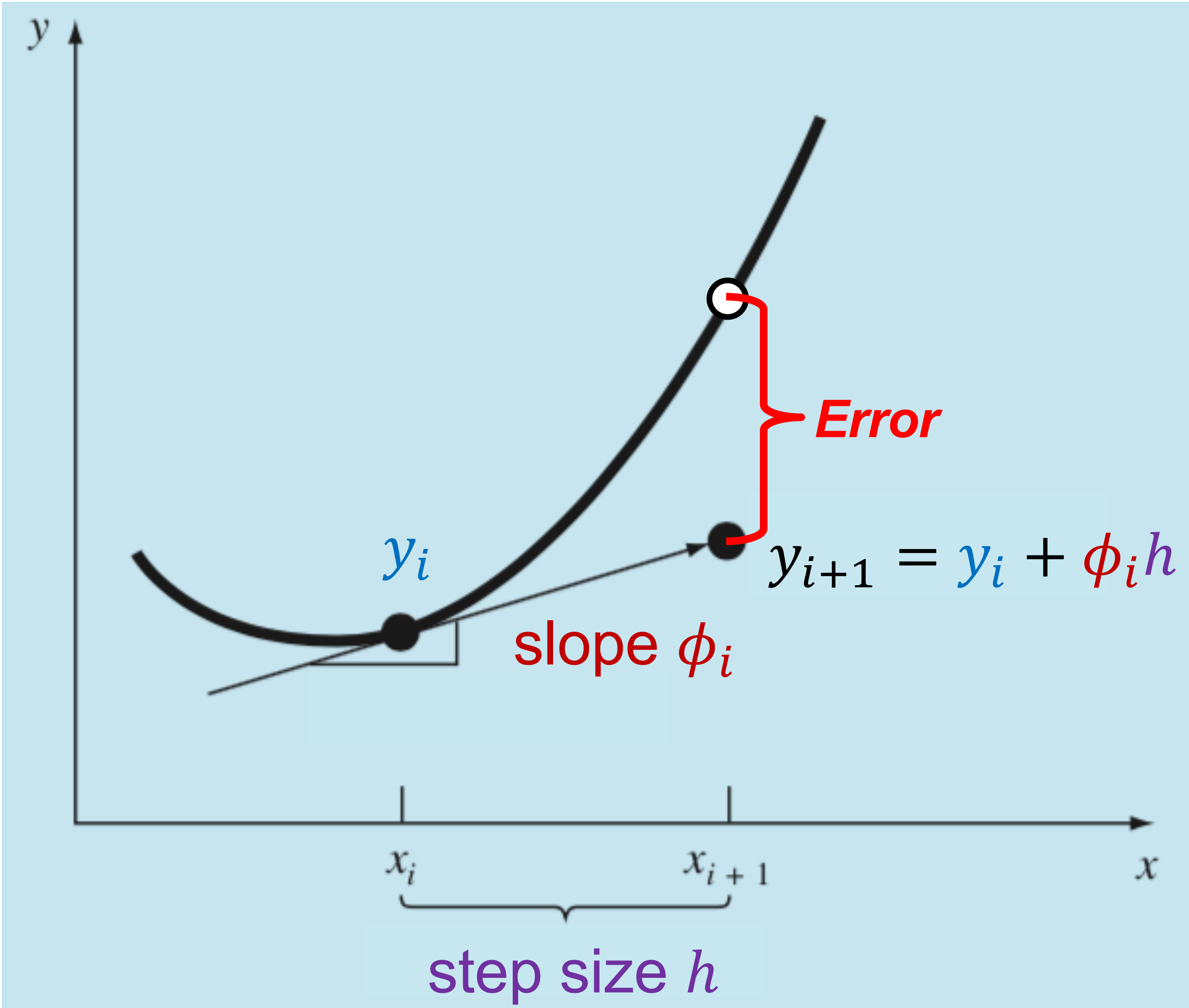
Single-step Methods

1) Start at $y(x_i) \equiv y_i$

2) Estimate slope ϕ_i

3) Take step of size h

4) Calculate y_{i+1} based on slope



Runge-Kutta Methods

- Accuracy of y_{i+1} obviously depends heavily on the choice of slope
  - For numerical differentiation we wanted to calculate unknown slope from known y_{i+1}

$$\phi_i = \frac{y_{i+1} - y_i}{h}$$

- These are the Runge-Kutta (RK) family of methods
  - Will start with **Euler's Method**, the simplest
  - Will build towards 4th-order RK, the most popular

Simplest RK: **Euler's Method**

- Governing ODE:

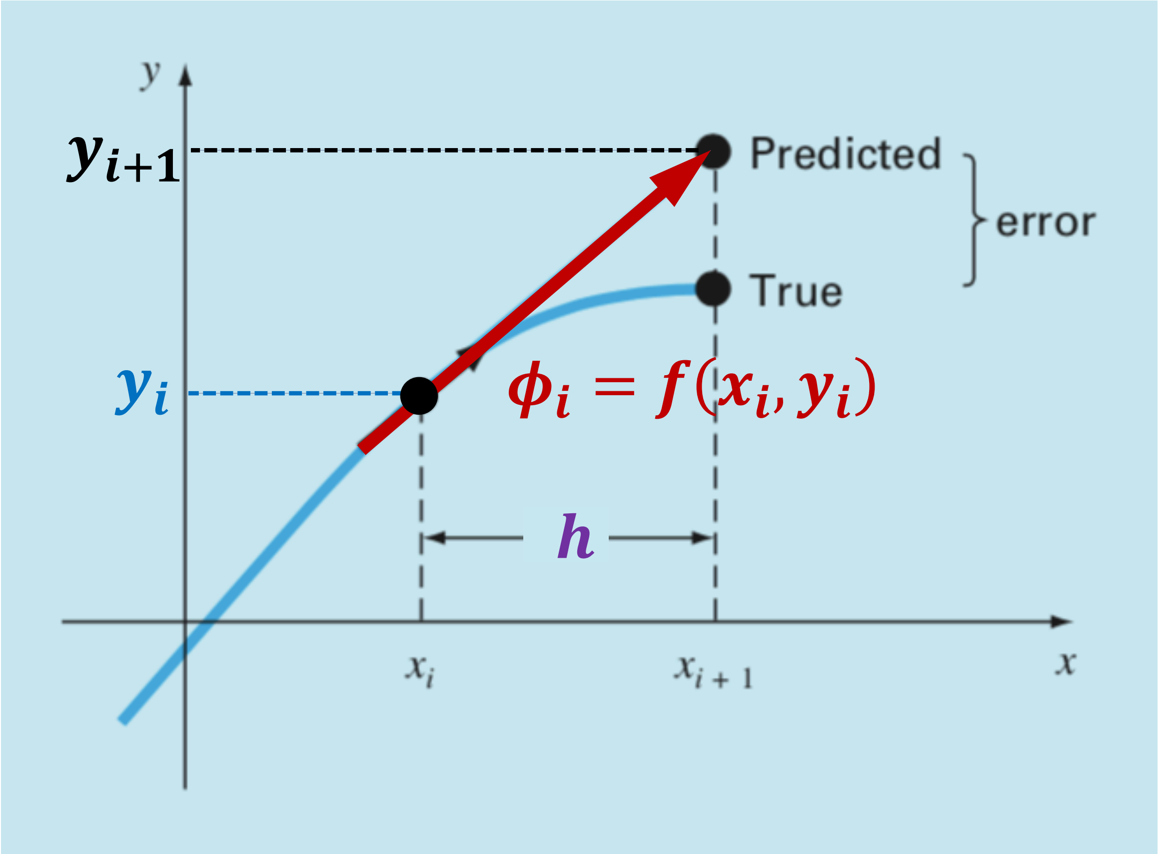
$$y' = f(x, y)$$

- Set ϕ = slope at i:

$$y_i' = f(x_i, y_i) \equiv \phi_i$$

- Euler **predictor**:

$$\begin{aligned} y_{i+1} &= y_i + \phi_i h \\ &= y_i + f(x_i, y_i)h \end{aligned}$$



- Smarter methods later, called **predictor-corrector**

Euler's Method: **Example**

- Solve for $y(x)$ using Euler's Method:

  $$\frac{dy}{dx} = -2x^3 + 12x^2 - 20x + 8.5 \quad [y' = f(x,y)]$$

- From $x = 0 \ldots 4$ to with $h = 0.5$
  - So number of steps = $\frac{(4-0)}{0.5} = 8$

- Initial condition: at $x = 0$, $y = 1$

  $$y(0) = 1$$

- $f(x,y)$ is a simple polynomial, so true solution available for error calculation:

  $$y(x) = 0.5x^4 + 4x^3 - 10x^2 + 8.5x + 1$$

Euler Example: **First step (h=0.5)**

$$y' = -2x^3 + 12x^2 - 20x + 8.5 \quad y(0) = 1$$

- Initial condition ($i = 0$)

  $x_i = x_0 = 0$

  $y(x_0) = y_0 = 1$

- Estimate slope $\phi_i = f(x_i, y_i)$ $[= y_i']$

  $\phi_0 = f(x_0, y_0) = f(0,1) = 8.5$

- Calculate $y_{i+1} = y_i + \phi_i h$

  $y_1 = y_0 + \phi_0 h = 1 + (8.5)(0.5) = \mathbf{5.25}$

  $x_1 = x_0 + h = 0 + 0.5 = \mathbf{0.5}$

Euler Example: **Second step (h=0.5)**

$$y' = -2x^3 + 12x^2 - 20x + 8.5 \quad y(0) = 1$$

- Previous step ($i = 1$)

  $x_1 = 0.5$

  $y(x_1) = y_1 = 5.25$

- Estimate slope $\phi_i = f(x_i, y_i)$ [$= y_i'$]

  $\phi_1 = f(x_1, y_1) = f(0.5, 5.25) = 1.25$

- Calculate $y_{i+1} = y_i + \phi_i h$

  $y_2 = y_1 + \phi_1 h = 5.25 + (1.25)(0.5) = \mathbf{5.875}$

  $x_2 = x_1 + h = 0.5 + 0.5 = \mathbf{1.0}$

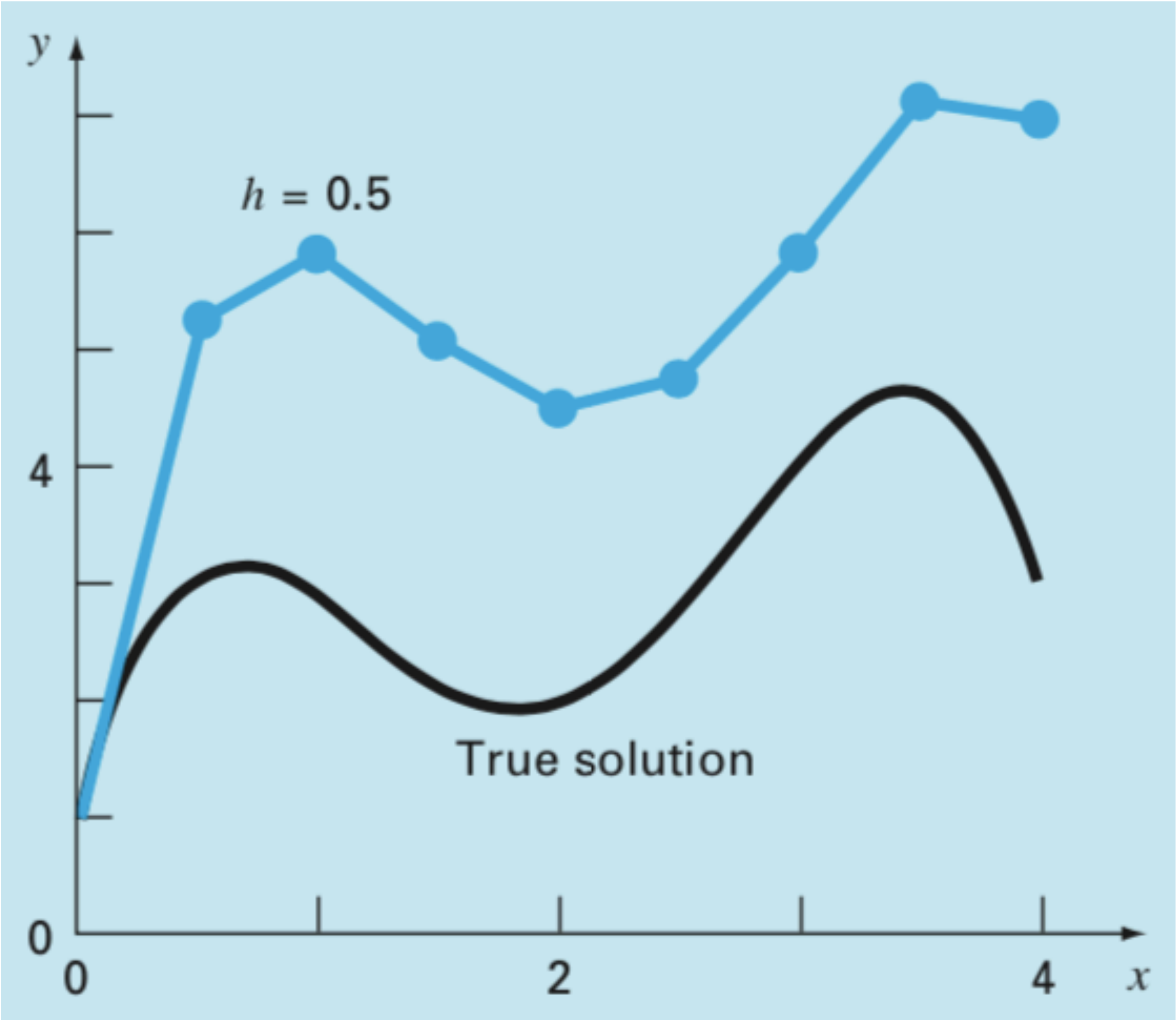
Euler Example: **Results (h=0.5)**

Initial condition →

Estimates of $y(x)$ for each x step

i	x_i	y_i	y_{true}	ε_t [%]
0	0.0	1.00000	1.00000	-
1	0.5	5.25000	3.21875	63%
2	1.0	5.87500	3.00000	96%
3	1.5	5.12500	2.21875	131%
4	2.0	4.50000	2.00000	125%
5	2.5	4.75000	2.71875	75%
6	3.0	5.87500	4.00000	47%
7	3.5	7.12500	4.71875	51%
8	4.0	7.00000	3.00000	133%

Euler Example: **Results (h=0.5)**



Euler's Method: **Error Analysis**

- Consider Taylor Series:

$$y_{i+1} = y_i + y_i' h + \frac{y_i'' h^2}{2!} + \frac{y_i''' h^3}{3!} + \cdots$$

- Remembering $y_i' = f(x_i, y_i)$:

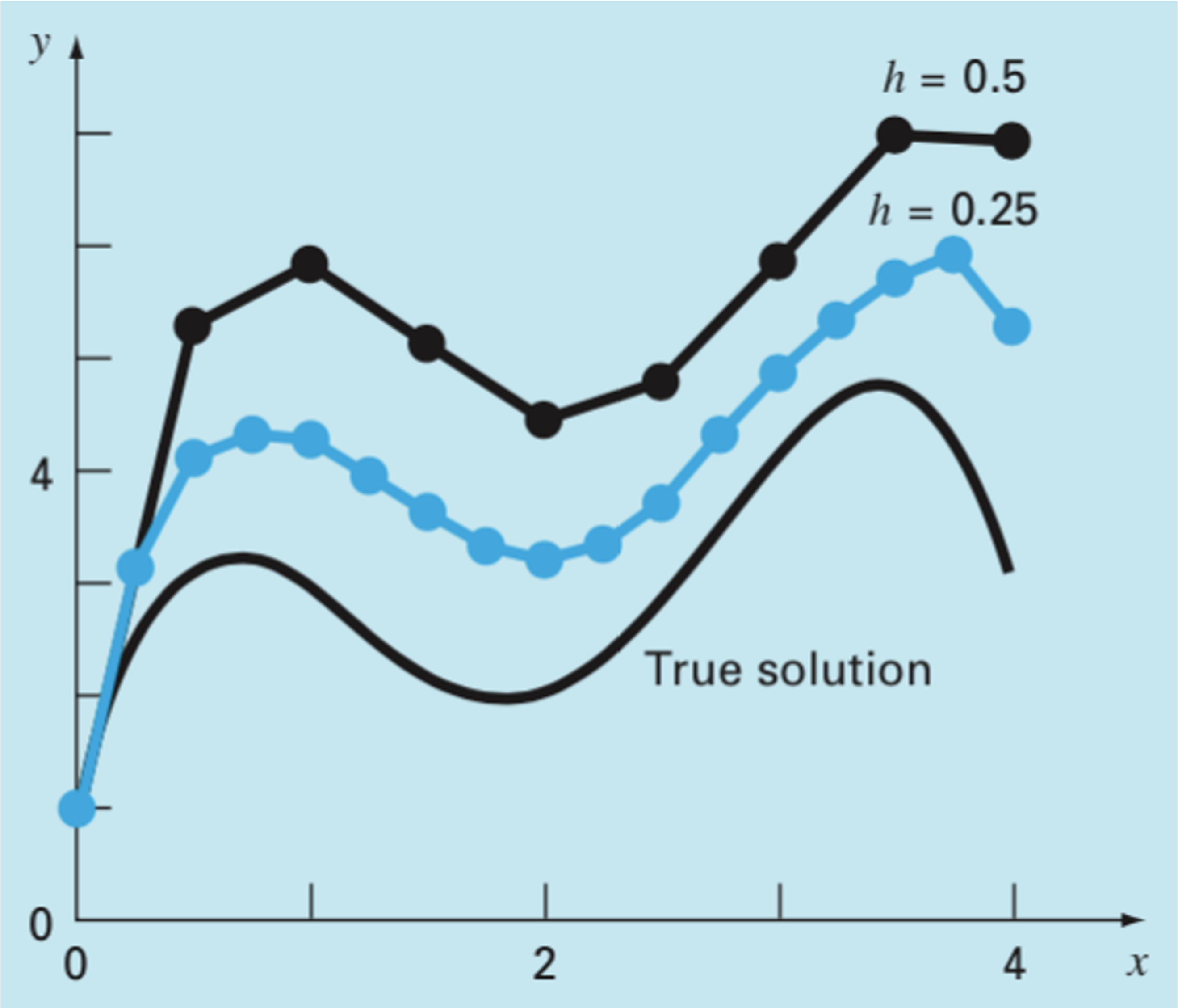
$$y_{i+1} = y_i + f(x_i, y_i)h + \frac{f'(x_i,y_i)h^2}{2!} + \frac{f''(x_i,y_i)h^3}{3!} + \cdots$$
$$= y_i + f(x_i, y_i)h + O(h^2)$$

- So error for Euler's Method should be $O(h^2)$
  - If we reduce h by factor of 2, error should drop 4x…
- Try $h = 0.25$ instead of $h = 0.5$
  - See **MIE334_Lecture_32_ExEuler.xlsx**

Euler Example: **Results (h=0.25 vs. h=0.5)**

i	x_i	y_i	y_{true} (h=0.25)	ε_t [%] (h=0.25)	ε_t [%] (h=0.5)	Error Drop
0	0.0	1.00000	1.00000	-	-	
2	0.5	4.17969	3.21875	30%	63%	2.1x
4	1.0	4.34375	3.00000	45%	96%	2.1x
6	1.5	3.55469	2.21875	60%	131%	2.2x
8	2.0	3.12500	2.00000	56%	125%	2.2x
10	2.5	3.61719	2.71875	33%	75%	2.3x
12	3.0	4.84375	4.00000	21%	47%	2.2x
14	3.5	5.86719	4.71875	24%	51%	2.1x
16	4.0	5.00000	3.00000	67%	133%	2.0x

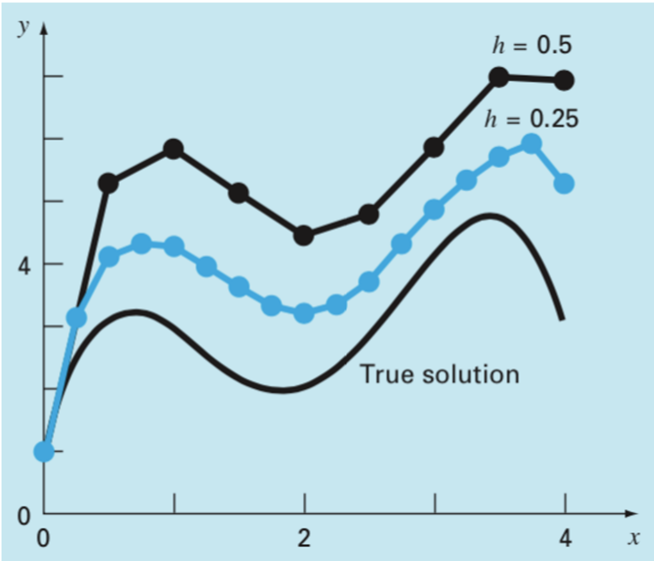
Euler Example: **Results (h=0.25 vs. h=0.5)**



Euler's Method: **Error Analysis**

- When solving ODEs, we have two types of error:
  - **Truncation**: nature of approximate mathematical techniques used to approximate values of function
  - **Roundoff**: finite precision (# of significant digits) in the representation of numbers

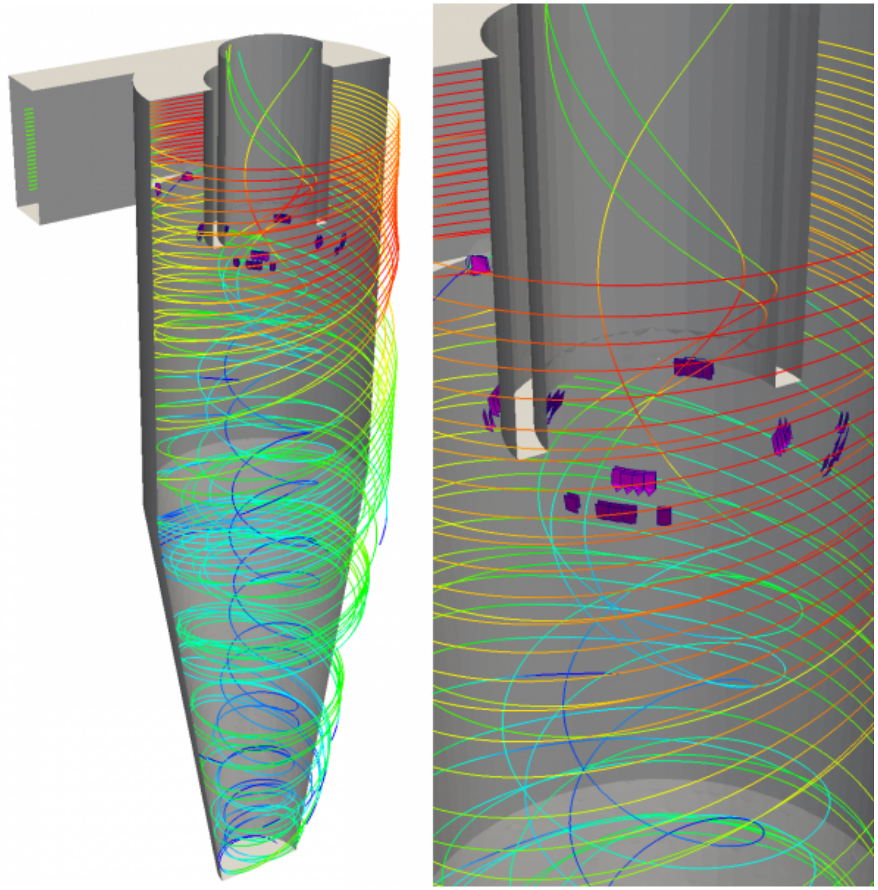
- **Truncation** error has **two** parts:
  - **Local**: error caused by truncating Taylor series
  - **Propagated**: previous errors can get us off track from true solution
  - Sum is the **Global** truncation error



CFD Particle Tracking: **Global Error**

- Velocity is a (time-varying) vector:
  - $\{u(t), v(t), w(t)\}$
- Compute pathlines by integrating each velocity component from initial "seed" locations:

$$dx/dt = u(x, y, z, t)$$
$$dy/dt = v(x, y, z, t)$$
$$dz/dt = w(x, y, z, t)$$

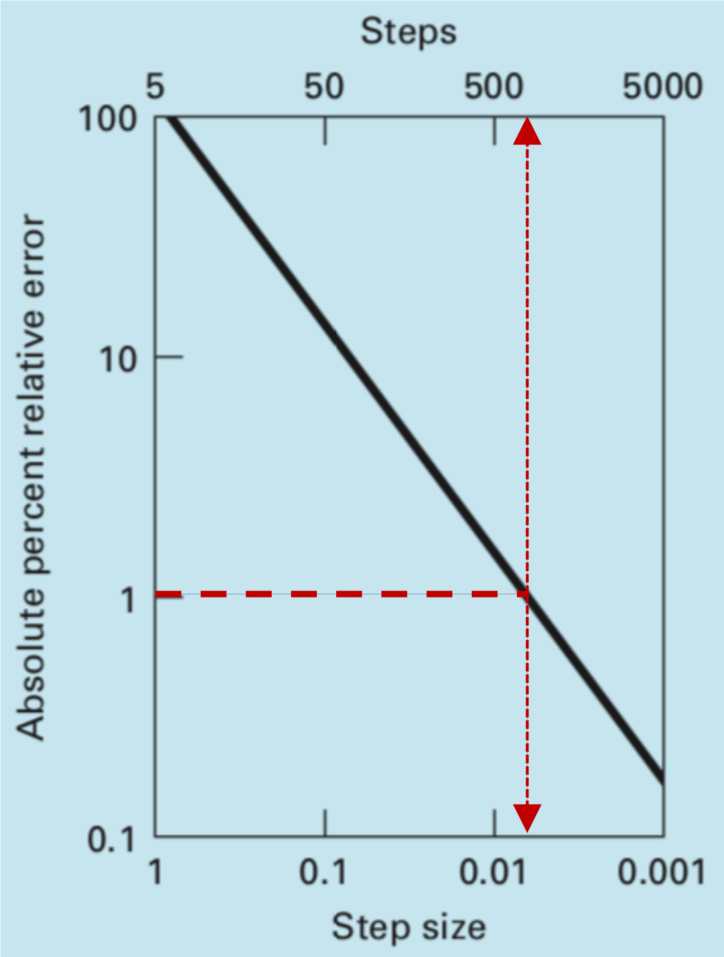


https://cfd.direct/openfoam/free-software/barycentric-tracking/

- Later will discuss numerical solution of multiple ODEs

Euler Method: **Limitations**

- Fundamental source of error in Euler method:
  - Local truncation error $O(h^2)$
  - Global truncation error $O(h)$
    - Because slope at **start** of interval assumed to apply across **entire** interval
- Obvious solution: reduce h:
  - $h = 0.5, \varepsilon_t = 133\%$, 8 steps
  - $h = 0.25, \varepsilon_t = 67\%$, 16 steps
- For $\varepsilon_s = 1.0\%$, would need more than 500 steps ($h < 0.01$)!



Higher-order Methods: **Multi-step**

- Euler Method: Uses 1st-order difference formula for y_i':

$$y_i' = \frac{y_{i+1} - y_i}{h} = f(x_i, y_i)\ [= \phi_i] \Rightarrow y_{i+1} = y_i + \phi_i h$$

- Could try 2nd-order difference formula for y_i':

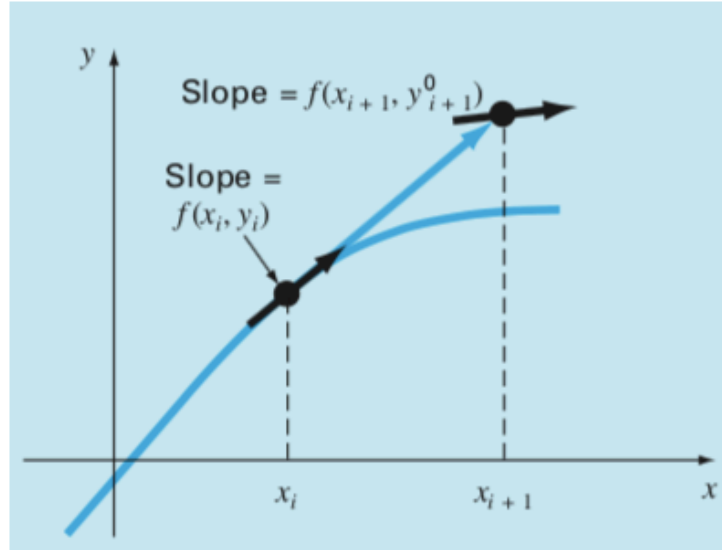
$$y_i' = \frac{-y_{i+2} + 4y_{i+1} - 3y_i}{2h} = \phi_i \ \Rightarrow\ y_{i+2} = 4y_{i+1} - 3y_i - 2\phi_i h$$

- Still needs to start with Euler method
    - **Propagated** error
- Still relies on slope from previous step
    - **Local** error
    - Clever ways around this (C&C 26, will skip)

Higher-order Methods: **Single step**

- Improve the estimation of the slope over the interval by using **more than one estimate** of the slope

- Heun's Method:
  - One at the **start**, one at the **end**



(a)

- Other methods later:
  - Different and/or more locations for slope estimation

Heun's Method

1) Slope at start (x_i):

$$k_1 = f(x_i, y_i)$$

2) Predictor (Euler):

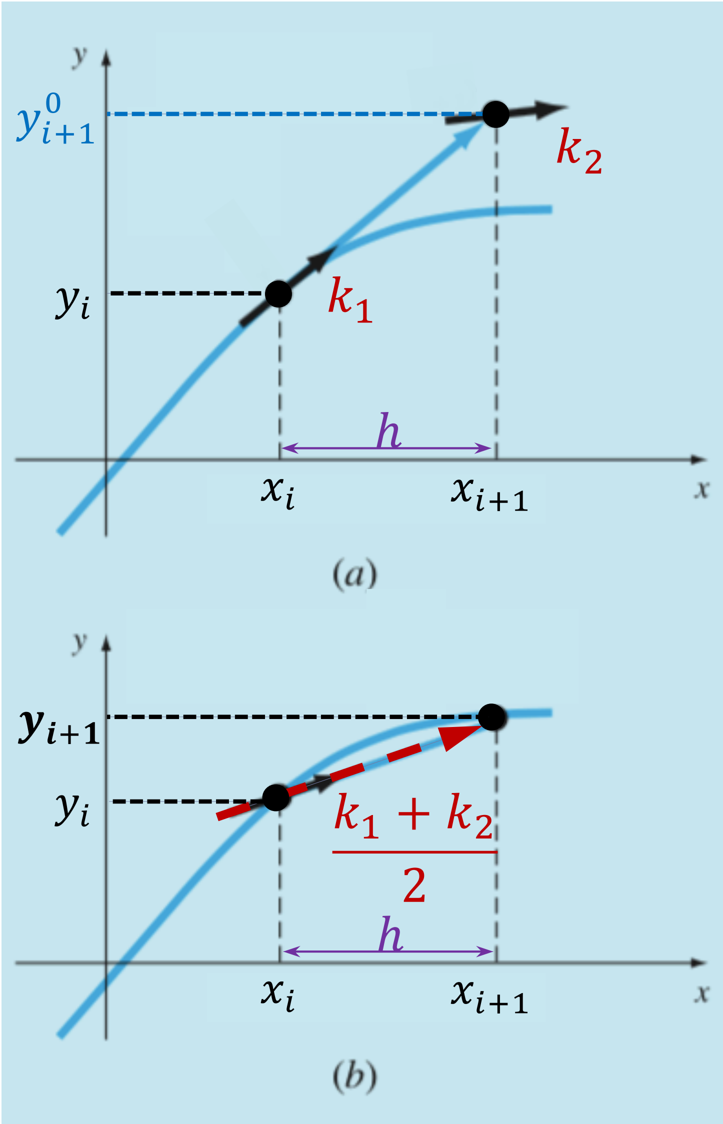
$$y_{i+1}^0 = y_i + k_1 h$$

3) Slope at end (x_{i+1}):

$$k_2 = f\left(x_{i+1}, y_{i+1}^0\right)$$

4) Corrector:

$$\boldsymbol{y_{i+1}} = y_i + \left(\frac{k_1 + k_2}{2}\right) h$$



Heun Example: **First step (h=0.5)**

$$y' = -2x^3 + 12x^2 - 20x + 8.5 \quad y(0) = 1$$

- Predictor:

$$k_1 = f(x_0, y_0) = f(0,1) = 8.5$$

$$y_1^0 = y_0 + k_1 h = 1 + (8.5)(0.5) = 5.25$$

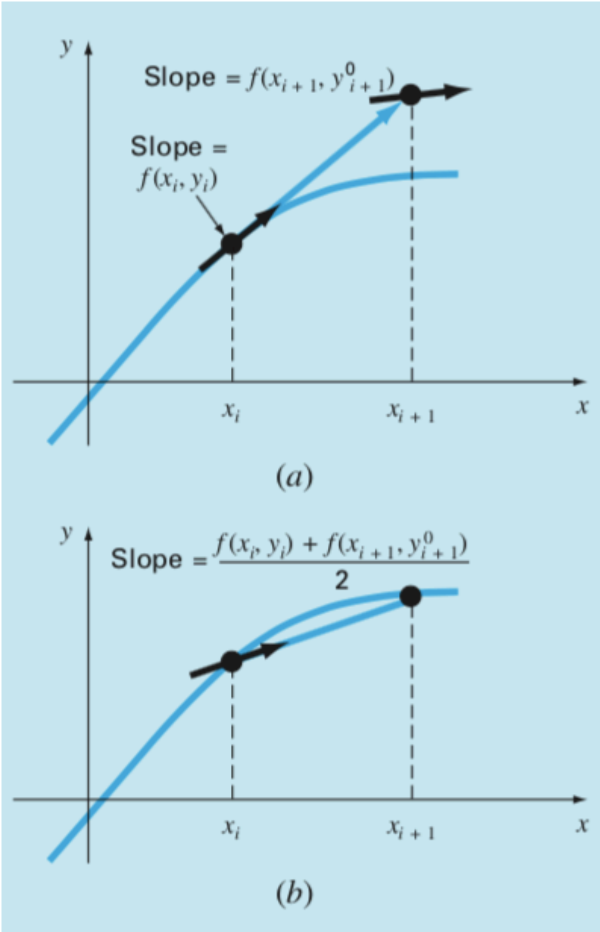
$$x_1 = x_0 + h = 0 + 0.5 = 0.5$$

- Corrector:

$$k_2 = f(x_1, y_1^0) = f(0.5, 5.25) = 1.25$$

$$y_1 = y_0 + \left(\frac{k_1 + k_2}{2}\right) h$$

$$= 1 + \left(\frac{8.5 + 1.25}{2}\right)(0.5) = \mathbf{3.4375}$$



Heun Example: **Second step (h=0.5)**

$$y' = -2x^3 + 12x^2 - 20x + 8.5 \quad y(0) = 1$$

- Predictor:

$$k_1 = f(x_1, y_1) = f(0.5, \mathbf{3.4375}) = 1.25$$

$$y_2^0 = y_{\cancel{0}\,1} + k_1 h = \mathbf{3.4375} + (1.25)(0.5) = 4.0625$$

$$x_2 = x_1 + h = 0.5 + 0.5 = 1.0$$

- Corrector:

$$k_2 = f(x_2, y_2^0) = f(1.0, 4.0625) = -1.5$$

$$y_2 = y_1 + \left(\frac{k_1 + k_2}{2}\right) h$$

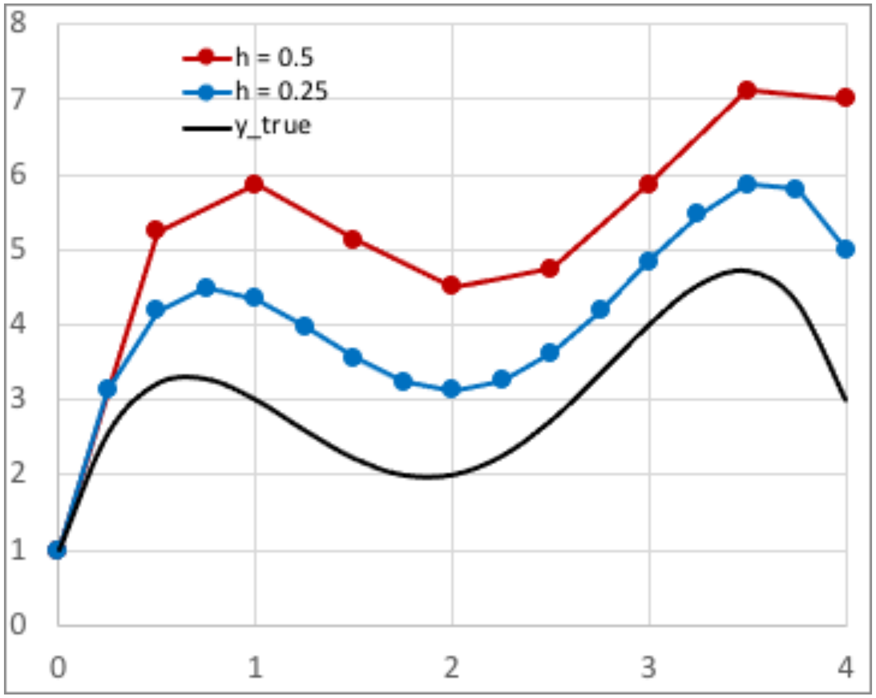
$$= \mathbf{3.4375} + \left(\frac{1.25 + (-1.5)}{2}\right)(0.5) = \mathbf{3.375}$$

Heun Example: **Results vs. Euler (h=0.5)**

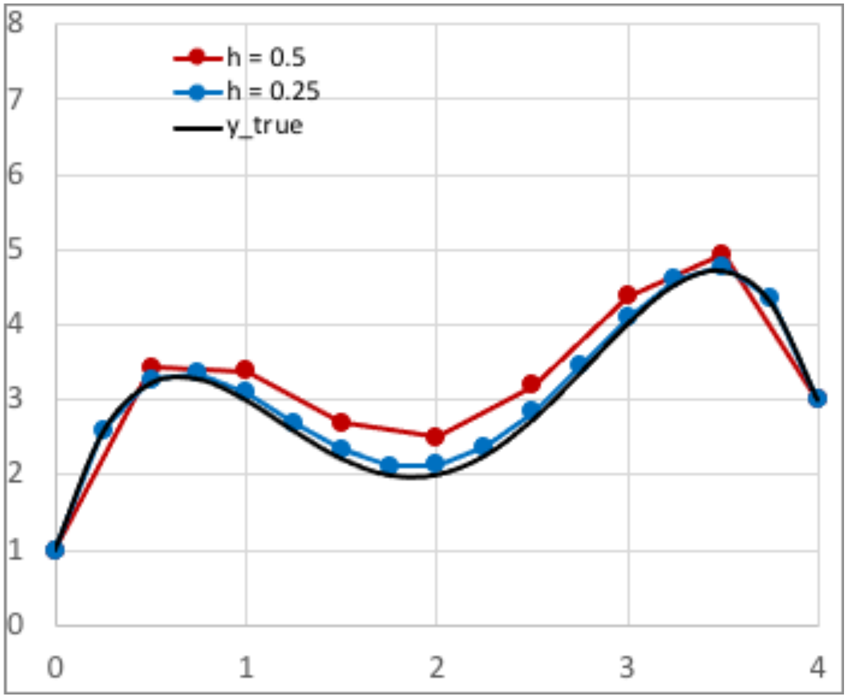
i	x_i	y_{Heun}	$\lvert\varepsilon_t\rvert$ [%]	y_{Euler}	$\lvert\varepsilon_t\rvert$ [%]	Error Drop
0	0.0	1	-	1.00000	-	-
1	0.5	3.43750	7%	5.25000	63%	9.0x
2	1.0	3.37500	13%	5.87500	96%	7.4x
3	1.5	2.68750	21%	5.12500	131%	6.2x
4	2.0	2.50000	25%	4.50000	125%	5.0x
5	2.5	3.18750	17%	4.75000	75%	4.4x
6	3.0	4.37500	9%	5.87500	47%	5.2x
7	3.5	4.93750	5%	7.12500	51%	10x
8	4.0	3.00000	0%	7.00000	133%	N/A

Heun Example: **Results vs. Euler**

Euler's Method



Heun's Method



MIE334_Lecture_32_ExEuler.xlsx

MIE334_Lecture_32_ExHeun.xlsx